Design Principles

Sean Barnum, Cigital, Inc. [vita[3]]
Michael Gegick, Cigital, Inc. [vita[4]]

2005-09-19

L4 / D/P, L[5]

As the recognition of security as a key dimension of high-quality software development has grown, the understanding of and ability to craft secure software has become a more common expectation of software developers. The challenge is in the learning curve. Most developers don't have the benefit of years and years of lessons learned that an expert in software security can call on. In an effort to bridge this gap, the Principles content area, along with the Guidelines and Coding Rules content areas, presents a set of practices derived from real-world experience that can help guide software developers in building more secure software.

Jerome Saltzer and Michael Schroeder were the first researchers to correlate and aggregate high-level security principles in the context of protection mechanisms [Saltzer 75]. Their work provides the foundation needed for designing and implementing secure software systems. Principles define effective practices that are applicable primarily to architecture-level software decisions and are recommended regardless of the platform or language of the software. As with many architectural decisions, the principles, which do not necessarily guarantee security, at times may exist in opposition to each other, so appropriate tradeoffs must be made. Software developers, whether they are crafting new software or evaluating and assessing existing software, should always apply these design principles as a guide and yardstick for making their software more secure.

The Principles content area presents several principles, many from Saltzer and Schroeder's original work and a couple of others from other thought leaders in the space. The filter applied to decide what is a principle and what is not is fairly narrow, recognizing that such lasting principles do not come along every day and that the term has been overused recently to define many things, causing confusion. Each principle consists of a brief description outlining the basic concept of the principle and then a set of more detailed descriptions in the form of block quotes from recognized thought leader publications describing their perspective on that particular principle. Rather than instigating conflict by acting as self-appointed arbiters in defining the one true interpretation of each principle, we decided to present readers with the different points of view available and allow them to make their own interpretations based on their personal trust filters. In doing this, editorial comment and explanatory prose has been kept to a minimum by design. It is our hope that readers of the principles, both expert and novice alike, will contribute to this explanatory discussion through the collaboration channels available on Build Security In. Eventually, these principles will likely be enhanced with content from those discussions.

The Principles for Software Security

- Securing the Weakest Link[7]
- Defense in Depth[8]
- Failing Securely[9]
- Least Privilege[10]
- Separation of Privilege[11]
- Economy of Mechanism[12]
- Least Common Mechanism[13]
- Reluctance to Trust[14]

[3] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)
[4] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/345-BSI.html (Gegick, Michael)

- Never Assuming that your Secrets are Safe[15]
- Complete Mediation[16]
- Psychological Acceptability[17]
- Promoting Privacy[18]

References

[Saltzer 75]	Saltzer, Jerome H. & Schroeder, Michael D. "The Protection of Information in Computer Systems," 1278-1308. *Proceedings of the IEEE 63*, 9 (September 1975).

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com